EXHIBIT 11

PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : G06F 17/00, 13/14, 9/46 — A1

(11) International Publication Number: WO 95/04974

(43) International Publication Date: 16 February 1995 (16.02.95)

(21) International Application Number: PCT/AU94/00454

(22) International Filing Date: 8 August 1994 (08.08.94)

(30) Priority Data:
PM 0416 6 August 1993 (06.08.93) AU

(71) Applicants *(for all designated States except US)*: THE COMMONWEALTH OF AUSTRALIA [AU/AU]; Anzac Park, Canberra, ACT 2600 (AU). FLINDERS TECHNOLOGIES PTY. LTD. [AU/AU]; 150 Grenfell Street, Adelaide, S.A. 5000 (AU).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: ASHMAN, Helen, Lee [AU/AU]; Defence Science and Technology Organisation, Information Technolgy Division, Commercial Road, Salisbury, S.A. 5108 (AU). VERBYLA, Janet [AU/AU]; Flinders Technologies Pty. Ltd., 150 Grenfell Street, Adelaide, S.A. 5000 (AU).

(74) Agent: R.K. MADDERN & ASSOCIATES; 345 King William Street, Adelaide, S.A. 5000 (AU).

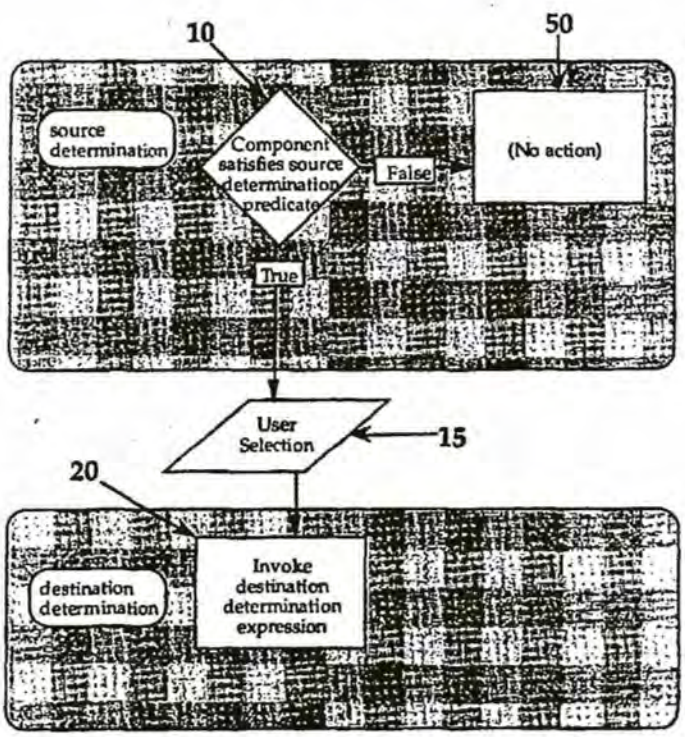
(81) Designated States: AM, AT, AU, BB, BG, BR, BY, CA, CH, CN, CZ, DE, DK, ES, FI, GB, GE, HU, JP, KE, KG, KP, KR, KZ, LK, LT, LU, LV, MD, MG, MN, MW, NL, NO, NZ, PL, PT, RO, RU, SD, SE, SI, SK, TJ, TT, UA, US, UZ, VN, European patent (AT, BE, CH, DE, DK, ES, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG), ARIPO patent (KE, MW, SD).

Published
With international search report.

(54) Title: A METHOD OF SPECIFYING LINKS IN HYPERMEDIA

(57) Abstract

This invention describes a method for implementing a fully dynamic and externalised linking strategy in a Hypermedia Management System (HMS), by using a (predicate, expression) representation for each link specification. This method improves on prior art because the predicate permits the HMS to determine whether a component is a source for a link, doing so in the most economic and efficient manner, avoiding all unnecessary computation and permitting automatic reconciliation of links with the underlying information collection if changes have occurred. In the figure, use of a predicate to represent the source determination process (10) is an essential step of the method of the invention. The next step, the destination determination process (20), is subject to the result of the source determination process, and is also subject to user selection (15). This invention involves the creation of links at the time they are required, as opposed to the usual practice of precomputing links. It also involves the use of link specifications, from which individual links are computed, replacing the need to specify individual links. These link specifications are abstracted out of the HMS application code, permitting them to be modified at any time, and permitting use of these link specifications by other HMSs.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	GB	United Kingdom	MR	Mauritania
AU	Australia	GE	Georgia	MW	Malawi
BB	Barbados	GN	Guinea	NE	Niger
BE	Belgium	GR	Greece	NL	Netherlands
BF	Burkina Faso	HU	Hungary	NO	Norway
BG	Bulgaria	IE	Ireland	NZ	New Zealand
BJ	Benin	IT	Italy	PL	Poland
BR	Brazil	JP	Japan	PT	Portugal
BY	Belarus	KE	Kenya	RO	Romania
CA	Canada	KG	Kyrgyzstan	RU	Russian Federation
CF	Central African Republic	KP	Democratic People's Republic of Korea	SD	Sudan
CG	Congo			SE	Sweden
CH	Switzerland	KR	Republic of Korea	SI	Slovenia
CI	Côte d'Ivoire	KZ	Kazakhstan	SK	Slovakia
CM	Cameroon	LI	Liechtenstein	SN	Senegal
CN	China	LK	Sri Lanka	TD	Chad
CS	Czechoslovakia	LU	Luxembourg	TG	Togo
CZ	Czech Republic	LV	Latvia	TJ	Tajikistan
DE	Germany	MC	Monaco	TT	Trinidad and Tobago
DK	Denmark	MD	Republic of Moldova	UA	Ukraine
ES	Spain	MG	Madagascar	US	United States of America
FI	Finland	ML	Mali	UZ	Uzbekistan
FR	France	MN	Mongolia	VN	Viet Nam
GA	Gabon				

A METHOD OF SPECIFYING LINKS IN HYPERMEDIA

This invention relates to Hypermedia Management Systems, and in particular a method of determining whether in a hypermedia management system there exists one or more links, and hence one or more destination components for a given source component using a (predicate, expression) representation of a link specification. Areas of application of the method include but are not limited to on-line information services, computer aided instruction, computer managed learning and multimedia information kiosks.

DEFINITIONS

To describe the limitations and problems of the prior art in hypermedia management systems, we first provide a few definitions (eg. Hypermedia Management System, Hypermedia System, Multimedia Information, information component, link, link specification and link generation). Next we explain how existing Hypermedia Management Systems function using these elements, and how the invention improves on the prior art.

In this document the following terms are used and their meanings for the purposes of this specification are as follows:

A *Hypermedia System* (HS) is a collection of information items, a means of displaying information items, and a means of navigating through the information collection.

Multimedia information is information which may exist in many media, for example, a graphic, text, sound, video each of various size.

A *component* is a discrete unit of information, i.e an information item.

A *link* represents an instance of a relationship between two or more components. The *source* component of a link is the component currently being viewed by a user, and the *destination* component(s) of a link are those components related to the source component and which the user can navigate to, by traversing the link.

A *link specification* is used to determine whether a link exists from the current (source) component, and may also, when required, be used to determine the destination(s).

It is possible to specify a relationship by a link specification without specifying or having to hand any instance of a link.

A *Hypermedia Management System* (HMS) typically comprises a software module that provides a set of functions for navigating through multimedia information, including (but not restricted to) the following features :

- determining what, if any, links apply to a given component;
- informing the application displaying the current (source) component that one or more links is available to be chosen by the user from the given component;
- determining the destination component(s) for a chosen link;
- invoking the appropriate software module or application which will display the destination endpoint(s) of the chosen link;
- the creation, modification and deletion of links and link specifications.

A *computed* link is created by evaluating a link specification to identify source components and corresponding destination component(s) to each source component. This is in contrast to a *manually-specified* link in which the link must be entered directly by a user or author, and with which there is no link specification associated.

A *dynamic* link is a computed link which is computed at the last possible moment, i.e. the computation is undertaken at the time the result of the computation is needed. This is in contrast to a *precomputed* link in which the link is computed and the results stored for later use.

A *predicate* is a function (in the mathematical sense) which takes a component (plus variables representing the environment in which the component occurs) and returns true or false.

An *expression* is a function which takes a component (plus variables representing the environment in which the component occurs) and returns a set of components (plus associated variables representing changes in the environment).

An example of a Hypermedia Management System is HyperCard, which consists of software designed for the Apple Macintosh computer to provide users with a Multimedia Information management tool. HyperCard also provides user control of movement between cards in a stack, typically by providing screen icons and buttons.

An example of a Hypermedia System is a HyperCard document, which consists of a series of "cards" (Multimedia Information components) associated with each other into a virtual card "stack"; each card can contain text, graphical images, and sound.

Note that the above definitions are not standard in the hypermedia community. Hypermedia is a new, emergent technology and as yet no standards of nomenclature and definitions have been agreed upon. Also, many HMS and HS designers tend to blur the distinctions given here, for example, many researchers and designers do not differentiate between computed links and dynamic links.

BACKGROUND

There are many different HMSs in current use, and there exists a large number of different HSs. HMSs can access many different information components which may be available in the HMS own internal memory devices or available from external sources (such as dial-up-databases and computer information storage devices as well as other HSs).

HMSs are intended to assist the initial creation of links by the original authors and co-ordinate to differing degrees the creation of the original links, maintenance of those links, the creation of new links by the author and the user, and the co-ordination of changes to existing links.

Typically, HMS designers make assumptions about the various information components it will be able to access, in regards to permanency, location, availability, format and access permissions (readability, writability). In doing so, designers accept a variety of limitations and compromises to the performance of the HMS. These compromises influence new link authorship, version management and propagation of editing changes to links involving the changed information and limitations which are acceptable in some situations.

Fundamental to every HMS is the management of links and, dependent on the type of assumptions inherent in the design of the HMS there will be an effect on the ability of a HMS to compute links upon request (referred to in this specification as the *dynamism* of the link), and the ability of the HMS to compute the link when certain elements required for the computation (including the link specifications themselves) are stored either internally or

externally of the HMS (referred to in this document as the *externalisation* of the link).

DYNAMISM

Links can be either manually entered by authors and/or users, or can be computed from link specifications. Computed links can be the result of either a precomputation of link specifications or a dynamic evaluation of link specifications. Some links can be a mixture of manually entered and computed. Likewise, links can be a mixture of precomputed and dynamic. This is because the source determination part of the link specification can be implemented differently to that of the destination determination part of the link specification.

In the prior art, sources of links are either manually-entered or precomputed. Destinations of links are either manually-entered, precomputed or dynamic. More recently, precomputed sources or destinations are tending to replace manually-entered sources or destinations.

Links which are computed will be accurate as long as the information which was provided to them at the time of computation does not change. However in some environments, the underlying information collection does change, either occasionally or frequently. When this occurs the links created earlier may become invalid. Reconciling links with the changes in the information collection is currently an important research area in the hypermedia research community.

When using links that have computed sources, designers tend to save and later re-use the results of those computations i.e. the link sources are precomputed, especially when there is a nontrivial amount of computation involved. Because of this, there is a higher likelihood that the underlying information collection will have changed between the time of computation and the time of use, causing inconsistencies in the precomputed links. It is possible to detect and correct such inconsistencies and a number of methods have been devised for this purpose. However, if both the source and destination(s) of each link are dynamically computed, then the inconsistencies never occur, and no corrective activities are needed.

Many of the "dynamic" linking techniques in the prior art are only partially dynamic, that is only the computation of the destination(s) is dynamic.

With a dynamic link, an individual link is computed every time it is required, so that accuracy of the results of the computation can be optimised. In the prior art, there is no HMS design allowing all links to be fully dynamic, that is, for links whose sources and destinations are all dynamic.

There are many situations and environments in which fully dynamic linking is preferable. Information collections which are subject to frequent change, which have very large collections of links, or whose accuracy is paramount, must be able to propagate those changes to the links immediately without intensive human input.

EXTERNALISATION

It is possible to identify links either by specifying individual pairs of components, or by giving link specifications which denote many pairs of components, derived from a parameterised computation. Note that it is possible to specify and store sets of individual pairs which are the result of a precomputation of a link specification.

If links are specified in individual pairs, then there are two possible ways to store these individual pairs. The first way is to embed special characters (tags) into the information collection which demarcate the components at the source and destination(s) of each link. This process is called "marking-up" the information collection.

The second way is to store references to the individual components in an external file, typically by naming the file containing the component and giving an offset within the file, that offset being where the component is located within the file. That is, the references to the individual components have been *externalised*. This second way is a core feature of the "open hypermedia systems" and an important advantage gained by this technique is that links do not have to be marked-up in the information collection, hence read-only information can be included. However this technique has also introduced a disadvantage, namely that editing changes in the information collection can make the link wrong. This problem now requires some corrective action in order to reconcile the links to the changed information collection. This problem does not occur when using the marking-up technique.

The alternative to specifying links in individual pairs is to use link specifications which compute individual links. This means that it is no longer

necessary to store individual pairs, but instead the link specifications now need to be stored. Link specifications can comprise two parts, the source determination and destination determination parts, but a separate source determination is not commonly used in the prior art.

Link specifications can be stored within the HMS application software, as part of the actual software code. A drawback of storing link specifications in this way is that the link specifications cannot be changed without altering the HMS application software code and recompiling it. When the link specifications refer to specific data structures, changes in these data structures can cause inconsistencies in the computed links, or can cause the HMS to stop working at all.

Alternatively, link specifications can be stored externally in a file (or other storage) from which the HMS retrieves them and then interprets them. These link specifications stored outside of the HMS application software code are *externalised*. Link specifications stored externally can be changed at any time, without the need to recompile.

It is possible to store the source determination part of the link specification and destination determination part of the link specification in different ways. In the prior art, sources are frequently stored individually, or in groups of individually named sources, often as the result of the precomputation of a source determination part of a link specification.

A preferred link specification is one in which both the source determination and the destination determination are externalised. Such a link specification has the following characteristics:

- the link specification computes demarcations of link source and destination components, hence there is no need to store such demarcations;
- link specifications are not part of the HMS application code but are stored in a form (such as a script) that the HMS can use;
- both the source determination and destination determination of each link specification are separately specified, so that there is no dependence upon internal HMS data structures for the successful operation of the link.

The benefits of full externalisation include:

- link specifications are stored where they may be useable by other HMSs;

- links may be specified in the languages proprietary to third-party applications, thus allowing information stored in these third-party applications to be accessed by the HMS, and allowing the HMS to make use of the specialist computational facilities of third-party applications;

- link specifications can be analysed by high-level intelligent applications, for the purposes of finding the 'best' representation of the link specification (link optimisation);

We have defined externalisation above as the separation of link specifications from the HMS software, and of the source and destination component demarcations from the information components. A consequence of this is that link specifications can be expressed in a language (such as Standard Query Language, Prolog etc) which is interpretable by some third-party application (such as Oracle, or a Prolog interpreter). In this way, it becomes possible to create link specifications which link information between third-party applications. This is called the *interoperability* of the HMS with third-party applications.

It is an aspect of the invention to provide a method which allows for full dynamism, full extensibility and interoperability where required but which will accommodate the prior forms of HSs and HMSs.

REPRESENTATIONS OF LINK SPECIFICATIONS

The fully dynamic linking principle depends on the replacement of individual links with link specifications. Link specifications are representations of groups of individual links as mathematical relations, and may be expressed as pairs of computation, one for each of the source and destination determinations. Link specifications need not be computed dynamically, however link specifications are a necessary (but not sufficient) condition for the implementation of fully dynamic links.

Any representation of a link specification as a mathematical relation must define the source set, the destination set and the instances (links) that comprise the relation (link specification). There are many possible ways to represent a mathematical relation, and for hypermedia purposes, some are better than others. However, fully dynamic linking can only be implemented when the source set is defined with a predicate.

The procedure implicit in almost all hypermedia link processes is the testing of whether a component (unit of information) currently on display is able to participate in a link.

That is, the process of using links consists of posing the question "given a particular component, what relationships does it participate in?".

This is in contrast to the querying/information retrieval approach to information delivery also used in HMSs in which the procedure is more like the question "given a relationship, find all components which participate in that relationship?". This corresponds to using an expression to determine the set of all sources for the link specifications.

The distinction is that the process of using a link requires a search for relationships (and as a result, other related components), whereas the query based retrieval requires a search for components. Hence a link is a form of context-sensitive query.

In brief, hypermedia link usage consists of asking "does this component have a link attached to it?" and if the answer is affirmative, then optionally requesting to be taken to the destination component(s) represented by the one or more links attached or associated with the component.

The prior art represents link specifications in a pair of set-creating computations, called *expressions*. Hence prior art link specifications consists of (expression, expression) pairs.

These (expression, expression) pairs can potentially be fully externalised (although this is not in general realised in the prior art), hence gaining many of the important advantages of externalisation.

However the full dynamism is not generally feasible using this representation of the link specification, because of the computational cost involved.

Also the (expression, expression) representation contradicts the basic process of hypermedia, that is, a user will be at a particular component and ask "what links are there from this component?". The (expression, expression) representation computes all of the sources for a particular link specification and then tests whether the component the user is interested in forms part of that set. But the user is not interested in what other components also belong to the same source set, they only want to know about the component they are currently viewing.

To compute the entire source set when only one component is likely to be used represents a predominantly redundant computation.

For operational purposes, there is no need whatsoever to compute the entire source set of the relation. The linking process does not require the entire source set to exist, but rather, it only requires that the current component can be tested for membership in that source set. This can be done with one predicate for each relationship (i.e. for each link specification).

The invention uses a (predicate, expression) representation of a link specification.

The predicate is a computation which comprises the necessary and sufficient conditions that the component must satisfy in order to be a source for the link specification. The set of sources for the link specification never becomes out of date because it is never fully computed. Only the component in which the user is interested is tested for whether it is a source for the link specification, and no redundant computation occurs at all.

In almost all cases, dynamically computing sources with a predicate provides significant computation savings. Only in very extreme cases is precomputation of the source set more economical computationally.

THE LINK SPECIFICATION MODEL

The following provides three important characteristics of the representation of a link specification:

- every link specification comprises a (predicate, expression) pair (P, E), where P and E are defined as follows:

  Let C be the set of components, and let ENV be the set of environments which apply to C.

  The set C is not restricted in any way by limitations of object size or type, but a component is essentially any discrete unit of information. The ENV set implicitly has an infinite number of dimensions, although most applications will only make use of a finite subspace of the ENV set in their computations (eg aspects such as navigational history, users preferences etc).